

(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 0 880 253 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
25.11.1998 Bulletin 1998/48

(51) Int Cl.⁶ **H04L 25/03**

(21) Application number: **98303697.1**

(22) Date of filing: **12.05.1998**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE**
Designated Extension States:
AL LT LV MK RO SI

(72) Inventors:
• **Werner, Jean-Jacques**
Holmdel, New Jersey 07733 (US)
• **Yang, Jian**
Ocean, New Jersey 07712 (US)

(30) Priority: **23.05.1997 US 862909**

(74) Representative: **Johnston, Kenneth Graham et al**
Lucent Technologies (UK) Ltd,
5 Mornington Road
Woodford Green Essex, IG8 OTU (GB)

(71) Applicant: **LUCENT TECHNOLOGIES INC.**
Murray Hill, New Jersey 07974-0636 (US)

(54) **Blind training of a decision feedback equalizer**

(57) A decision feedback equalizer (DFE) comprises a feed-forward filter and a feedback filter. Blind training of the DFE is performed using a statistical-based tap

updating algorithm for the feed-forward filter, and a symbol-based type of tap updating algorithm for the feedback filter.

EP 0 880 253 A2

Best Available Copy

Description**Cross-Reference to Related Applications**

Related subject matter is disclosed in the following co-pending, commonly assigned, U.S. Patent applications of Werner et al.: serial No. 08/646404, filed on May 7, 1996; serial No. 08/717582, filed on September 18, 1996; serial No. 08/744908, filed on November 8, 1996; serial No. 08/757207, filed on November 27, 1996; and serial No. 08/766406, filed on December 12, 1996.

Background Of the Invention

The present invention relates to communications equipment, and, more particularly, to blind equalization in a receiver having a decision feedback equalizer (DFE).

In blind equalization, the linear equalizer of a receiver is converged without the use of a training signal. As known in the art, there are two techniques for blind equalization: one is referred to herein as the "reduced constellation algorithm" (RCA) (e.g., see Y. Sato, "A Method of Self-Recovering Equalization for Multilevel Amplitude-Modulation Systems," *IEEE Trans. Commun.*, pp. 679 - 682, June 1975; and U.S. Patent No. 4,227,152, issued October 7, 1980 to Godard); and the other technique is the so-called "constant modulus algorithm" (CMA) (e.g., see D. N. Godard, "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communications Systems," *IEEE Trans. Commun.*, vol. 28, no. 11, pp. 1867-1875, Nov. 1980; and N. K. Jablon, "Joint Blind Equalization, Carrier Recovery, and Timing Recovery for High-Order QAM Signal Constellations," *IEEE Trans. Signal Processing*, vol. 40, no. 6, pp. 1383-1398, 1992.) Further, the co-pending, commonly assigned, U.S. Patent application of Werner et al., entitled "Blind Equalization," serial No. 08/646404, filed on May 7, 1996, presents a new blind equalization technique - the multimodulus algorithm (MMA) - as an alternative to the above-mentioned RCA and CMA approaches.

However, in applications where the dominant noise includes one or several radio frequency (RF) interferers, a linear equalizer alone is not able to provide good performance. In such applications, it is desirable to use a decision feedback equalizer (DFE). A DFE comprises a feed-forward portion and a feedback portion. The latter is used to correct postcursor interference.

Unfortunately the above-described approaches to blind equalization are not well-suited to the blind equalization of a DFE.

Summary of the Invention

We have realized that the key to blindly converge a DFE is to use different types of tap updating algorithms for the feed-forward and feedback portions. In particular, the above-mentioned blind equalization techniques reduce the probability of error in a statistical sense. However, a DFE requires the actual, and hopefully correct, symbols to be fed back to the feedback portion of the DFE in order to correct the postcursor interference. If the correct symbols are not fed back to the feedback filter, then error propagation occurs and both the feed-forward and feedback filter will not converge properly. Therefore, and in accordance with the inventive concept, a statistical-based tap updating technique is applied to the feed-forward portion and a symbol-based tap updating technique is applied to the feed-back portion.

In an embodiment of the invention, a DFE comprises a feed-forward filter and a feed-back filter. An RCA tap updating algorithm is used to blindly converge the feed-forward filter and a symbol-sliced least mean squared (LMS) tap updating algorithm is used to converge the feed-back filter.

In another embodiment of the invention, a DFE comprises a feed-forward filter and a feed-back filter. An MMA tap updating algorithm is used to blindly converge the feed-forward filter and a generalized LMS tap updating algorithm is used to converge the feed-back filter.

Brief Description of the Drawing

FIG. 1 is an illustrative block diagram of a prior art communications system using a linear equalizer;
 FIG. 2 is an illustrative block diagram of a prior art communications system using a decision feedback equalizer;
 FIG. 3 is an illustrative impulse response showing precursors and postcursors;
 FIG. 4 is a block diagram of a prior art two-dimensional DFE;
 FIG. 5 is a block diagram of a prior art one-dimensional DFE during blind start-up using the RCA technique;
 FIG. 6 is an illustrative flow chart embodying the principles of the invention; and
 FIGs. 7 - 10 show illustrative embodiments in accordance with the principles of the invention;
 FIG. 11 is a block diagram of a prior art phase-splitting noise predictive decision feedback equalizer (NPDFE);
 FIGs. 12 - 13 show other illustrative embodiments in accordance with the principles of the invention for use with

an NPDFE-type equalizer; and

FIGs. 14 and 15 are illustrative block diagrams of a portion of a receiver embodying the principles of the invention.

Detailed Description

Before describing the inventive concept, some background information is presented on decision feedback equalizers (DFEs). Also, generally speaking, equalizers operate in two modes - a training mode and a steady-state mode. During steady state, the LMS algorithm is typically used. During training, there can either be a training signal or training can be performed blind. The description below relates to blind training of a DFE.

Decision Feedback Equalizers

A typical transceiver structure is shown in FIG. 1. For simplicity, this figure and FIG. 2 show a one-dimensional transmission scheme. However, the discussion equally applies to the in-phase or quadrature signal path of a two-dimensional scheme such as CAP (Carrierless Amplitude/Phase Modulation). FIG. 1 shows a model of a communications system 10 that is disturbed by some noise $\xi(t)$. This communications system comprises shaping filter 15, communications channel 20, and linear equalizer 50. The latter comprises adaptive filter 25 and slicer 30. As known in the art, the error signal e_n is used to adaptively update tap coefficients (not shown) of linear equalizer 50. In many Local Area Network (LAN), Digital Subscriber Line (DSL), and other applications, sometimes the noise, $\xi(t)$, includes one or several dominant radio frequency (RF) interferers. Unfortunately, the presence of significant RF interference degrades the performance of a receiver using a linear equalizer only.

As known in the art, the effect of RF interference is mitigated when a feedback filter is used in the equalizer. As such, in order to improve the equalization performance with RF interference, use of DFEs have been proposed by those skilled in the art. As used herein, a typical DFE is called a conventional DFE (CDFE). FIG. 2 shows an illustrative structure of a CDFE, 90, using a feed-forward filter 60 and a feedback filter 70.

In both FIGs. 1 and 2, $w(t)$ is defined as the overall impulse response of the signal path so that:

$$w(t) = s(t) \otimes h(t) \otimes c(t), \quad (1)$$

where \otimes denotes convolution, $s(t)$, $h(t)$ and $c(t)$ are the impulse responses of the shaping filter, the channel, and the filter of the equalizer, respectively. For the linear equalizer of FIG. 1, the equalizer output signal $y(t)$ is written as

$$y(t) = a(t) \otimes w(t). \quad (2)$$

In FIG. 1, $a(t)$ represents a sequence of real symbols, a_n . If the output signal $y(t)$ is sampled at time instants $t = nT$, then

$$y_n = a_n w_0 + \sum_{m<0} a_{n-m} w_m + \sum_{m>0} a_{n-m} w_m + \xi_n. \quad (3)$$

At the sampling instant nT in equation (3), the first term represents the desired symbol, the second term involves the precursor channel symbols, the third term involves the postcursor symbols, and the last term ξ_n represents an additive noise. FIG. 3 shows an interpretation of the right side of equation (3) in the time-domain of an impulse response. In the case of FIG. 2, where the equalizer incorporates a feedback filter, equation (3) is written as:

$$y_n = y_{n,f} - z_n = y_{n,f} - \sum_{m>0} \hat{a}_{n-m} w_m = a_n w_0 + \sum_{m<0} a_{n-m} w_m + \sum_{m>0} (a_{n-m} - \hat{a}_{n-m}) w_m + \xi_n, \quad (4)$$

where \hat{a}_n is a sliced symbol. Equation (4) presents a different view of the equalizer, where the output signal of the equalizer, y_n , is the subtraction of the output signal, $y_{n,f}$, of the feed-forward filter and the output signal, z_n , of the feedback filter. The right side of equation (4) means that if the prior decisions are correct, i.e., the third term $\sum_{m>0} (a_{n-m} - \hat{a}_{n-m}) w_m$ is equal to zero, the output signal, y_n , of the equalizer just needs to be determined by the current symbol

$a_n w_0$, the precursor ISI, and the noise term ξ_n .

Now, consider a cost function, as known in the art, of a CDFE. For analysis purposes, a two-dimensional CDFE, 100, which uses a phase-splitting filter structure is shown in FIG. 4. CDFE 100 comprises analog-to-digital (A/D) converter 105, inphase finite impulse response (FIR) filter 110, quadrature phase FIR filter 150, symbol-slicers 115 and 155, feedback filter 120, 125, 160 and 165, and adders 130, 135, 170, and 175. CDFE 100 minimizes the following cost function:

$$CF = E \left[|Y_n - \hat{A}_n|^2 \right], \quad (5)$$

where \hat{A}_n is the complex output signal of the slicer. As shown in FIG. 4, the output signal of the equalizer, Y_n , consists of two components $Y_{f,n}$ and Z_n .

$$Y_n = Y_{f,n} - Z_n, \quad (6)$$

where $Y_n = y_n + j\tilde{y}_n$, $Y_{f,n} = y_{f,n} + j\tilde{y}_{f,n}$, and $Z_n = z_n + j\tilde{z}_n$. So that the input signal of the slicer, Y_n , is the subtraction of the output signal, $Y_{f,n}$, of the feed-forward filter and the output signal, Z_n , of the feedback filter. The vectors of input signals to the feedback filter are defined as:

$$\hat{\mathbf{a}}_n^T = [\hat{a}_{n-1}, \dots, \hat{a}_{n-k+1}], \quad (7)$$

and

$$\hat{\mathbf{b}}_n^T = [\hat{b}_{n-1}, \dots, \hat{b}_{n-k+1}]. \quad (8)$$

Note, that the input signals of the feedback filter are delayed versions of the sliced symbols. Because there are two vectors of input signals, which are $\hat{\mathbf{a}}_n$ and $\hat{\mathbf{b}}_n$, and two feedback filter, which are $\hat{\mathbf{b}}_{1,n}$ and $\hat{\mathbf{b}}_{2,n}$, the following outputs result:

$$z_n = \mathbf{b}_{1,n}^T \hat{\mathbf{a}}_n + \mathbf{b}_{2,n}^T \hat{\mathbf{b}}_n, \quad (9)$$

and

$$\tilde{z}_n = \mathbf{b}_{1,n}^T \hat{\mathbf{b}}_n - \mathbf{b}_{2,n}^T \hat{\mathbf{a}}_n. \quad (10)$$

The cost function in equation (5) is suitable for the LMS algorithm. This means that good performance is achieved when correct decisions, as represented by \hat{A}_n , are provided.

Filter Adaptation Algorithm for a CDFE

In this section, the steady-state filter adaptation algorithms for a CDFE with the phase splitting structure shown in FIG. 4 are derived. The gradients of the cost function in equation (5) with respect to the feed-forward filter vectors \mathbf{c}_n and \mathbf{d}_n are equal to:

$$\nabla_{\mathbf{c}} = 2E[e_{r,n} \mathbf{r}_n], \quad (11)$$

and

$$\nabla_d = 2E[\tilde{e}_n \mathbf{r}_n]. \quad (12)$$

Using these two equations in a stochastic gradient algorithm, the following tap updating algorithms for the LMS algorithm are obtained

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu \mathbf{e}_n \mathbf{r}_n = \mathbf{c}_n - \mu (\mathbf{y}_n - \hat{\mathbf{a}}_n) \mathbf{r}_n, \quad (13)$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu \tilde{\mathbf{e}}_n \mathbf{r}_n = \mathbf{d}_n - \mu (\tilde{\mathbf{y}}_n - \hat{\mathbf{b}}_n) \mathbf{r}_n, \quad (14)$$

where μ is the step size for the feed-forward filter. The gradients of the cost function in equation (5) with respect to the feedback tap vectors $\mathbf{b}_{1,n}$ and $\mathbf{b}_{2,n}$ are equal to:

$$\nabla_{b_1} = -2E[e_n \hat{\mathbf{a}}_n + \tilde{\mathbf{e}}_n \hat{\mathbf{b}}_n], \text{ and} \quad (15)$$

$$\nabla_{b_2} = -2E[e_n \hat{\mathbf{b}}_n - \tilde{\mathbf{e}}_n \hat{\mathbf{a}}_n]. \quad (16)$$

The following tap updating algorithms for the feedback filter result:

$$\mathbf{b}_{1,n+1} = \mathbf{b}_{1,n} + \mu_b [e_n \hat{\mathbf{a}}_n + \tilde{\mathbf{e}}_n \hat{\mathbf{b}}_n], \quad (17)$$

$$\mathbf{b}_{2,n+1} = \mathbf{b}_{2,n} + \mu_b [e_n \hat{\mathbf{b}}_n - \tilde{\mathbf{e}}_n \hat{\mathbf{a}}_n], \quad (18)$$

where μ_b is the step size of the feedback filter, and

$$\mathbf{e}_n = \mathbf{y}_n - \hat{\mathbf{a}}_n \quad \tilde{\mathbf{e}}_n = \tilde{\mathbf{y}}_n - \hat{\mathbf{b}}_n. \quad (19)$$

Problems with Blind CDFE

As known in the art, it is possible to achieve promising steady-state performance with a CDFE, but one cannot guarantee good results during blind start-up. The condition to make a feedback filter converge is to make $\sum_{m=0}^{\infty} (\mathbf{a}_{n-m} - \hat{\mathbf{a}}_{n-m})$ converge to zero in equation (4), which can only be done when the decisions are correct. For a CDFE, this condition cannot be satisfied during blind start-up.

For example, a CDFE filter structure during blind start-up using an RCA blind start-up approach is shown in FIG. 5. (It should be noted, that complex notation is used on the signal paths of FIGs. 5, 7, 8, 9, 10, 12, and 13, to represent either one-dimensional or two-dimensional signals, e.g., $\mathbf{Y}_{f,n}$). As is shown in FIG. 5, during blind start-up the RCA blind equalization algorithm is used for both the feed-forward filter 905 and the feed-back filter 915. A 4-point slicer, 910, is used to generate input symbols to feed-back filter 915. RCA uses the 4-point slicer to reduce the probability of wrong decisions in a statistical sense.

Using the RCA blind start-up approach, equation (4) becomes:

$$y_n - \sum_{m>0} R \operatorname{sgn}(y_n) w_0 = a_n w_0 + \sum_{m<0} a_{n-m} w_m + \sum_{m>0} (a_{n-m} - R \operatorname{sgn}(y_n)) w_m + \xi_n, \quad (20)$$

where R is a constant used in the RCA approach (e.g., see the above-mentioned co-pending U. S. Patent application of Werner et al., serial No. 08/646404, filed on May 7, 1996, for a detailed description of various blind-startup approaches). As a result, referring back to FIG. 4, the cost function for the two-dimensional structure is now given by:

$$CF = E[(Y_n - R \operatorname{sgn}(Y_n))^2], \quad (21)$$

where the output signal, Y_n , of the equalizer is defined in equation (6).

For a CDFE which directly uses the RCA blind algorithm, the decision symbol \hat{a}_n in equation (4) is changed to the term $R \operatorname{sgn}(y_n)$ in equation (20). By doing this, the term $\sum_{m>0} (a_{n-m} - R \operatorname{sgn}(y_n)) w_m$ in equation (20) clearly does not converge to zero when more than two values are used for the symbols a_{n-m} . As noted above, the 4-point slicer 910 reduces the probability of wrong decisions in a statistical sense. This helps feed-forward filter 905, but not the feedback filter 915, because the latter requires the correct input signals in order to cancel the post-cursors. If the correct symbols are not fed to feedback filter 915, then error propagation occurs and both feed-forward 905 and feedback filter 915 will not converge properly.

Algorithmic Structures of a Symbol-Sliced DFE

In accordance with the inventive concept, the blind start-up difficulties of a DFE are overcome if the feed-forward filter and the feedback filter are trained separately during blind start-up. An illustrative method is shown in FIG. 6.

A blind start-up procedure using a transition algorithm can be schedule-driven, event driven, or both. With a schedule-driven approach, the switch between two different tap updating algorithms occurs after some fixed number, M , of iterations (which can be determined by a counter, for example). With an event-driven approach, the switch occurs when a certain quality of eye opening is achieved. This can be done, for example, by continuously monitoring the MSE and making the switch when the MSE is below some threshold T . Values for M and T depend on the application and are determined experimentally. Illustratively, FIG. 6 shows a schedule-driven approach (an event-driven approach is similar and will not be described).

In step 205, the feed-forward filter is trained using a statistical-based, or blind equalization algorithm, such as RCA or MMA, which is known to work for a linear equalizer and uses statistical knowledge of the symbols. However, in step 210, the feedback filter is trained with an algorithm, such as LMS, which minimizes a cost function that is defined on the actual symbols of the signal constellation and does not use statistical knowledge of the symbols, i.e., is symbol-based.

As shown in step 215, the SDFE algorithm is used until the number of iterations, n , is greater than M . Once this condition is reached, the transition algorithm switches to using the LMS algorithm in step 230. Finally when the eye opens even more, e.g., to an MSE less than or equal to T , the receiver switches to a steady-state mode.

In accordance with the principles of the invention, a new one-dimensional CDFE is shown in FIG. 7, where a blind RCA algorithm for feed-forward filter 955 and the LMS algorithm for the feedback filter 965 are combined. With this new CDFE algorithm, the feed-forward filter 955 opens the eye with RCA, and the feedback filter 965 performs post-cursor cancellation with the LMS algorithm.

In general, the DFE in FIG. 7 during start-up is referred to herein as a symbol-sliced DFE (SDFE) because a symbol slicer is used for the feedback filter during training. This SDFE approach improves convergence performance during blind start-up. Comparing FIG. 5 and FIG. 7, the input signals to the feedback filter and the tap updating algorithms for the feedback filter are different. For the CDFE of FIG. 5, the symbols $\hat{A}_{r,n}$ are the input signals of feedback filter 915, where $\hat{A}_{r,n}$ are the outputs of 4-point slicer 910. In other words, in FIG. 5, RCA is used for the feedback filter. In contrast, for the SDFE of FIG. 7, the input signals to feedback filter 965 are the output symbols, \hat{A}_n , of symbol slicer 960. For a 16-CAP system, for example, this is a 16-point slicer. In FIG. 7, the LMS algorithm is used for feedback filter 965. Thus, and in accordance with the inventive concept, the feed-forward filter operates with a blind algorithm and the feedback filter operates with the LMS algorithm or symbol-based algorithm.

In accordance with the principles of the invention, several types of blind equalization algorithms can be used during blind start-up, such as RCA, CMA, and MMA. RCA is a commonly used blind equalization algorithm because it has the simplest implementation. CMA is reliable but is expensive due to the use of a rotator. MMA is a new blind algorithm that was originally proposed in above-mentioned co-pending U. S. Patent application of Werner et al. MMA provides a

good compromise between cost and performance, especially for the phase-splitting filter structure.

SDFE-RCA

The SDFE in FIG. 7 is also referred to herein as an SDFE-RCA because it uses RCA as the blind equalization technique for the feed-forward filter. In FIG. 7, RCA minimizes the following cost function for the feed-forward filter:

$$CF_f = E[|Y_n - A_{f,n}|^2] = E[|Y_n - R \operatorname{sgn}(Y_n)|^2], \quad (22)$$

where CF_f refers to the cost function for the feed-forward path. The vectors of symbol-sliced inputs used for the feedback filter are given by:

$$\hat{a}_n^T = [\hat{a}_{n-1}, \dots, \hat{a}_{n-k+1}], \quad (23)$$

and

$$\hat{b}_n^T = [\hat{b}_{n-1}, \dots, \hat{b}_{n-k+1}]. \quad (24)$$

Because the input signals to the feedback filter are different from those of the blind CDFE, a new cost function must be developed. As shown in FIG. 7, the LMS algorithm is used as the tap adaptation algorithm for the feedback filter. As known in the art, the cost function of the LMS algorithm is:

$$CF_b = E[|Y_n - \hat{A}_n|^2], \quad (25)$$

where CF_b refers to the cost function for the feedback filter.

SDFE-MMA

Turning to FIG. 8, another variation of an SDFE is shown, herein referred to as an SDFE-MMA in which the MMA blind equalization algorithm is used for the feed-forward filter during blind start-up. FIG. 8 shows the block diagram of an SDFE-MMA filter structure. With MMA, the algorithm minimizes the following cost function:

$$CF_f = E[(y_n^2 - R^2)^2 + (\tilde{y}_n^2 - R^2)^2], \quad (26)$$

where $R^2 = \frac{E[a_n^4]}{E[a_n^2]}$ (A detailed study of MMA is found in the above-mentioned co-pending U.S. patent application of Werner et al.) Like SDFE-RCA, the inputs to the feedback filter are the outputs of the symbol slicer, as defined in equations (23) and (24). However, the tap adaptation algorithm for the feedback filter is different from SDFE-RCA. Unlike RCA, which is a second-order algorithm, MMA is a fourth-order statistical algorithm. However, LMS is a second-order algorithm. In order to achieve similar convergence rates, it is desirable to adjust the LMS algorithm to a fourth-order algorithm. The cost function for the feedback filter is correspondingly adjusted to a fourth-order cost function referring to the sliced symbols. In accordance with the inventive concept, it is given by:

$$CF_b = E[(y_n^2 - \hat{a}_n^2)^2 + (\tilde{y}_n^2 - \hat{b}_n^2)^2]. \quad (27)$$

As used herein, equation (27) is referred to as a generalized fourth-order LMS algorithm (GLMS-4). As shown in FIG. 8, MMA is used for the feed-forward filter and the GLMS-4 algorithm is used for the feedback filter.

Other embodiments of the inventive concept are possible, as illustrated in FIGs. 9 and 10. In FIG. 9, RCA is used for the feed-forward filter and the GLMS-4 for the feedback filter, and in FIG. 10, MMA is used for the feed-forward filter and the LMS algorithm for the feedback filter. FIGs. 9 and 10 provide alternative embodiments that are easier to

implement. However, SDFE-RCA and SDFE-MMA shown in FIGs. 7 and 8 are recommended because of convergence rate considerations mentioned earlier. (Note, if CMA is used for the feed-forward filter, e.g., in FIG. 10, a rotator must be added after the feed-forward filter).

The inventive concept is equally applicable to noise predictive decision feedback equalizers (NPDFE). As described further below, a new NPDFE with sliced symbols for the feedback filter is referred to herein as an SNPDFE. For simplicity, only the cost functions and filter adaptation algorithms are provided below.

SNPDFE-RCA

For reference purposes, FIG. 11 shows a prior art NPDFE that uses a phase-splitting structure. In accordance with the invention, FIG. 12 shows the structure of a SNPDFE second-order algorithm, referred to herein as SNPDFE-RCA. The cost function for the feed-forward filter in FIG. 12 is as follows:

$$CF_f = E[|Y_{f,n} - R \text{sgn}(y_{f,n})|^2]. \quad (28)$$

The inputs to the feedback filter are given by the subtraction between the equalizer outputs $Y_{f,n}$ and slicer outputs A_n :

$$\hat{e}_{f,n}^T = [\hat{a}_{n-1} - y_{n-1}, \dots, \hat{a}_{n-k-1} - y_{n-k-1}], \quad (29)$$

and

$$\hat{e}_{q,n}^T = [\hat{b}_{n-1} - \tilde{y}_{n-1}, \dots, \hat{b}_{n-k-1} - \tilde{y}_{n-k-1}]. \quad (30)$$

The cost function for the feedback filter is the same as given in equation (25). With the input vectors defined in equations (29) and (30), the input signals of the slicer in equation (25) are given by:

$$y_n = y_{f,n} - z_n = y_{f,n} - (b_{1,n} e_{f,n} + b_{2,n} e_{q,n}), \quad (31)$$

and

$$\tilde{y}_n = \tilde{y}_{f,n} - \tilde{z}_n = \tilde{y}_{f,n} - (b_{1,n} e_{f,n} - b_{2,n} e_{q,n}) \quad (32)$$

SNPDFE-MMA

FIG. 13 shows a structure of an SNPDFE fourth-order algorithm, referred to herein as SNPDFE-MMA. The input signals to the feedback filter are the same as those for SNPDFE-RCA which are given in equations (29) and (30). However, the cost function for both the feed-forward filter and the feedback filter are different. For the feed-forward filter, the cost function is:

$$CF_f = E[(y_{f,n}^2 - R^2)^2 + (\tilde{y}_{f,n}^2 - R^2)^2]. \quad (33)$$

The cost function for the feedback filter is the same as that for SDFE-MMA defined in equation (27) with the input signals to the feedback filter given in equations (29) and (30) and the input signals of the slicer calculated in equations (31) and (32).

Filter Adaptation

In this section, the tap updating algorithms are presented. For simplicity, only two variations of SDFE are described.

SDFE-RCA

For SDFE-RCA, the standard RCA blind equalization technique is used for the feed-forward filter. The cost function of RCA is shown in equation (22) and the tap updating algorithms for the in-phase tap vector \mathbf{c}_n and quadrature tap vector \mathbf{d}_n are:

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu(y_n - R \operatorname{sgn}(y_n))\mathbf{r}_n, \quad (34)$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu(\tilde{y}_n - R \operatorname{sgn}(\tilde{y}_n))\mathbf{r}_n, \quad (35)$$

The cost functions of SDFE is different from that of blind CDFE. For the feedback filter, the cost function of SDFE is defined in equation (25) that is the same as the LMS algorithm. The tap updating algorithms are given by:

$$\mathbf{b}_{1,n+1} = \mathbf{b}_{1,n} + \mu_b((y_n - \hat{a}_n)\hat{\mathbf{a}}_n + (\tilde{y}_n - \hat{b}_n)\hat{\mathbf{b}}_n), \quad (36)$$

$$\mathbf{b}_{2,n+1} = \mathbf{b}_{2,n} + \mu_b((y_n - \hat{a}_n)\hat{\mathbf{b}}_n - (\tilde{y}_n - \hat{b}_n)\hat{\mathbf{a}}_n), \quad (37)$$

where the vectors $\hat{\mathbf{a}}_n$ and $\hat{\mathbf{b}}_n$ are defined in equations (23) and (24).

SDFE-MMA

The tap updating algorithms for SDFE-MMA are now derived. As described above, an illustrative structure for an SDFE-MMA is shown in FIG. 8. Again, MMA is used for the feed-forward filter. The tap updating algorithms for the MMA cost function are given by:

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu y_n(y_n^2 - R^2)\mathbf{r}_n, \quad (38)$$

and

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu \tilde{y}_n(\tilde{y}_n^2 - R^2)\mathbf{r}_n. \quad (39)$$

With SDFE-MMA, the input signals to the feedback filter are the same as those used for SDFE-RCA. However, in order to make the convergence rate consistent in the system, the LMS algorithm is modified to a fourth-order LMS algorithm, as given in equation (27). The gradients of the cost function in equation (27) with respect to the feedback tap vectors $\mathbf{b}_{1,n}$ and $\mathbf{b}_{2,n}$ are equal to:

$$\frac{\partial F_b}{\partial \mathbf{b}_1} = -4E[(y_n^2 - \hat{a}_n^2)y_n\hat{\mathbf{a}}_n + (\tilde{y}_n - \hat{b}_n^2)\tilde{y}_n\hat{\mathbf{b}}_n], \text{ and} \quad (40)$$

$$\frac{\partial F_b}{\partial \mathbf{b}_2} = -4E[(y_n^2 - \hat{a}_n^2)y_n \hat{\mathbf{b}}_n - (\tilde{y}_n^2 - \hat{b}_n^2)\tilde{y}_n \hat{\mathbf{a}}_n]. \quad (41)$$

The taps are updated in the opposite direction of the gradients. As a result, the following tap updating algorithms are obtained:

$$\mathbf{b}_{1,n+1} = \mathbf{b}_{1,n} + \mu_b[(y_n^2 - \hat{a}_n^2)y_n \hat{\mathbf{a}}_n + (\tilde{y}_n^2 - \hat{b}_n^2)\tilde{y}_n \hat{\mathbf{b}}_n], \text{ and} \quad (42)$$

$$\mathbf{b}_{2,n+1} = \mathbf{b}_{2,n} + \mu_b[(y_n^2 - \hat{a}_n^2)y_n \hat{\mathbf{b}}_n - (\tilde{y}_n^2 - \hat{b}_n^2)\tilde{y}_n \hat{\mathbf{a}}_n]. \quad (43)$$

SNPDFE-RCA

For the cost function in equation (28), the filter adaptation algorithms for the feed-forward filter are derived as:

$$c_{n+1} = c_n - \mu(y_{f,n} - R \operatorname{sgn}(y_{f,n}))r_n, \quad (44)$$

and

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu(\tilde{y}_{f,n} - R \operatorname{sgn}(\tilde{y}_{f,n}))r_n. \quad (45)$$

For the cost function of equation (25) and definitions shown in equations (31) and (32), the adaptation algorithms for the feedback filter are the following:

$$\mathbf{b}_{1,n+1} = \mathbf{b}_{1,n} + \mu_b((y_n - \hat{a}_n)\hat{\mathbf{e}}_{i,n} + (\tilde{y}_n - \hat{b}_n)\hat{\mathbf{e}}_{q,n}), \text{ and} \quad (46)$$

$$\mathbf{b}_{2,n+1} = \mathbf{b}_{2,n} + \mu_b((\tilde{y}_n - \hat{b}_n)\hat{\mathbf{e}}_{i,n} - (y_n - \hat{a}_n)\hat{\mathbf{e}}_{q,n}), \quad (47)$$

where the vectors $\hat{\mathbf{e}}_{i,n}$ and $\hat{\mathbf{e}}_{q,n}$ are defined in equations (29) and (30).

SNPDFE-MMA

For the cost function shown in equation (33), the filter adaptation algorithms for the feed-forward filter are given as:

$$c_{n+1} = c_n - \mu y_{f,n}(y_{f,n}^2 - R^2)r_n, \quad (48)$$

and

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu \tilde{y}_{f,n}(\tilde{y}_{f,n}^2 - R^2)r_n. \quad (49)$$

For the cost function shown in equation (27) and the definitions shown in equations (31) and (32), the updating

algorithms for the feedback filter are derived as:

$$\mathbf{b}_{1,n+1} = \mathbf{b}_{1,n} + \mu_b[(y_n^2 - \hat{a}_n^2)y_n \hat{\mathbf{e}}_{i,n} + (\tilde{y}_n^2 - \hat{b}_n^2)\tilde{y}_n \hat{\mathbf{e}}_{q,n}], \text{ and} \quad (50)$$

$$\mathbf{b}_{2,n+1} = \mathbf{b}_{2,n} + \mu_b[(\tilde{y}_n^2 - \hat{b}_n^2)\tilde{y}_n \hat{\mathbf{e}}_{i,n} - (y_n^2 - \hat{a}_n^2)y_n \hat{\mathbf{e}}_{q,n}]. \quad (51)$$

As described above, a new DFE algorithm, referred to herein as symbol-sliced DFE (SDFE), is used to improve blind equalization. With SDFE, the feed-forward filter uses a blind equalization algorithm and the feedback filter uses a symbol-based algorithm. The combined use of blind and symbol-based algorithm leads to an improvement in blind equalization.

Illustrative embodiments of the inventive concept are shown in FIGs. 14 and 15. FIG. 14 illustrates an embodiment representative of a digital signal processor 400 that is programmed to implement a DFE in accordance with the principles of the invention. Digital signal processor 400 comprises a central processing unit (processor) 405 and memory 410. A portion of memory 410 is used to store program instructions that, when executed by processor 405, implement the SDFE-type operation. This portion of memory is shown as 411. Another portion of memory, 412, is used to store tap coefficient values that are updated by processor 405 in accordance with the inventive concept. It is assumed that a received signal 404 is applied to processor 405, which equalizes this signal in accordance with the inventive concept to provide a output signal 406. For the purposes of example only, it is assumed that output signal 406 represents a sequence of output samples of a decision feedback equalizer. (As known in the art, a digital signal processor may, additionally, further process received signal 404 before deriving output signal 406.) An illustrative software program is not described herein since, after learning of the inventive concept as illustrated by the flow chart of FIG. 6, such a program is within the capability of one skilled in the art. Also, it should be noted that any equalizer structures, such as those described earlier, can be implemented by digital signal processor 400 in accordance with the inventive concept.

FIG. 15 illustrates another alternative embodiment of the inventive concept. Circuitry 500 comprises a central processing unit (processor) 505, and an equalizer 510. The latter is illustratively assumed to be a DFE. It is assumed that equalizer 510 includes at least two tap-coefficient register for storing values for corresponding tap coefficient vectors in the feed-forward and feedback fillers. Processor 505 includes memory, not shown, similar to memory 410 of FIG. 14 for implementing the SDFE-type algorithms. Equalizer output signal 511, which represents a sequence of equalizer output samples, is applied to processor 505. The latter analyzes equalizer output signal 511, in accordance with the inventive concept, to adapt values of the tap coefficients in such a way as to converge to a correct solution.

The foregoing merely illustrates the principles of the invention and it will thus be appreciated that those skilled in the art will be able to devise numerous alternative arrangements which, although not explicitly described herein, embody the principles of the invention and are within its spirit and scope.

For example, although the invention is illustrated herein as being implemented with discrete functional building blocks, e.g., an equalizer, etc., the functions of any one or more of those building blocks can be carried out using one or more appropriate programmed processors.

Claims

1. A method for converging an equalizer having a feed-forward portion and a feedback portion, the method comprising the steps of:
 - using a statistical-based equalization technique for converging the feed-forward portion and
 - using a symbol-based equalization technique for converging the feedback portion.
2. The method of claim 1 wherein the statistical-based equalization technique is based on either a reduced constellation algorithm (RCA), or on a constant modulus algorithm (MMA), or on a multimodulus algorithm (MMA).
3. The method of claim 1 wherein the symbol-based equalization technique is based on a least-mean square (LMS) algorithm.
4. The method of claim 1 wherein the symbol-based equalization technique is based on a fourth-order least-mean square (LMS) algorithm.

5. Apparatus comprising:

a feedback filter;

a feed-forward filter;

an N-symbol slicer operative on an applied signal, which is developed as a function of output signals from the feedback filter and the feed-forward filter, for generating a first sequence of symbols taken from a constellation comprising N symbols;

an M-symbol slicer operative on the applied signal for generating a second sequence of symbols taken from a constellation having M symbols, where $N < M$;

wherein the feed-forward filter adapts as a function of values of the first sequence of symbols and the feedback filter adapts as a function of values of the second sequence of symbols.

6. The apparatus of claim 5 wherein the feed-forward filter adapts as a function of a blind equalization algorithm and the feedback filter adapts as a function of a symbol-based algorithm.

7. The apparatus of claim 6 wherein the symbol-based algorithm is a least-mean square (LMS)-based algorithm.

8. The apparatus of claim 6, wherein the symbol-based algorithm is a fourth-order least-mean square (LMS)-based algorithm.

9. The apparatus of claim 5, wherein the feed-forward filter and the feedback filter form a decision feedback equalizer.

10. The apparatus of claim 5 wherein the feed-forward filter and the feedback filter form a noise predictive decision feedback equalizer.

FIG. 1 PRIOR ART

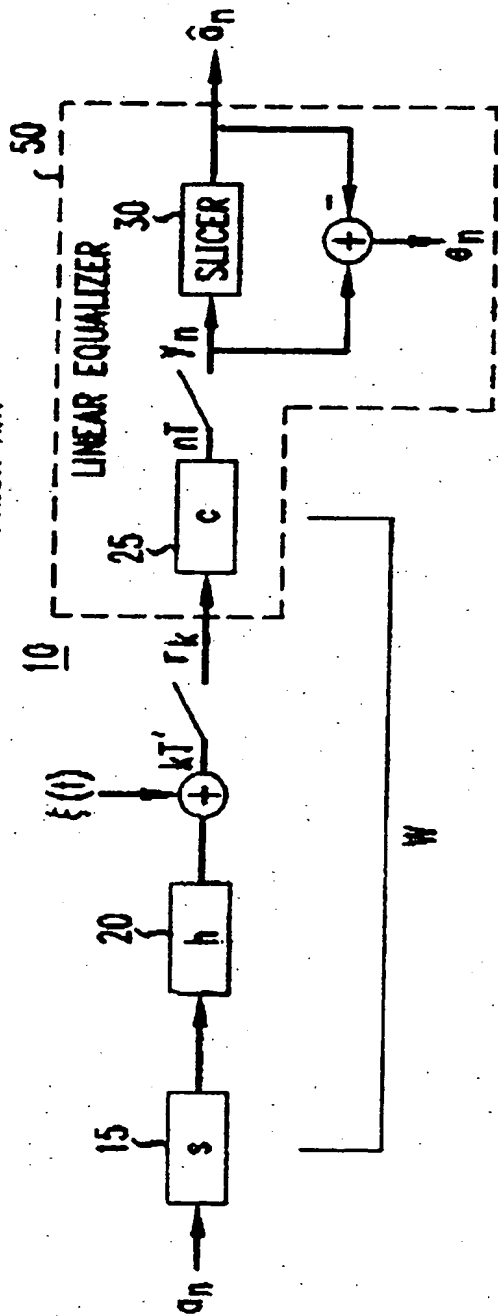


FIG. 2 PRIOR ART

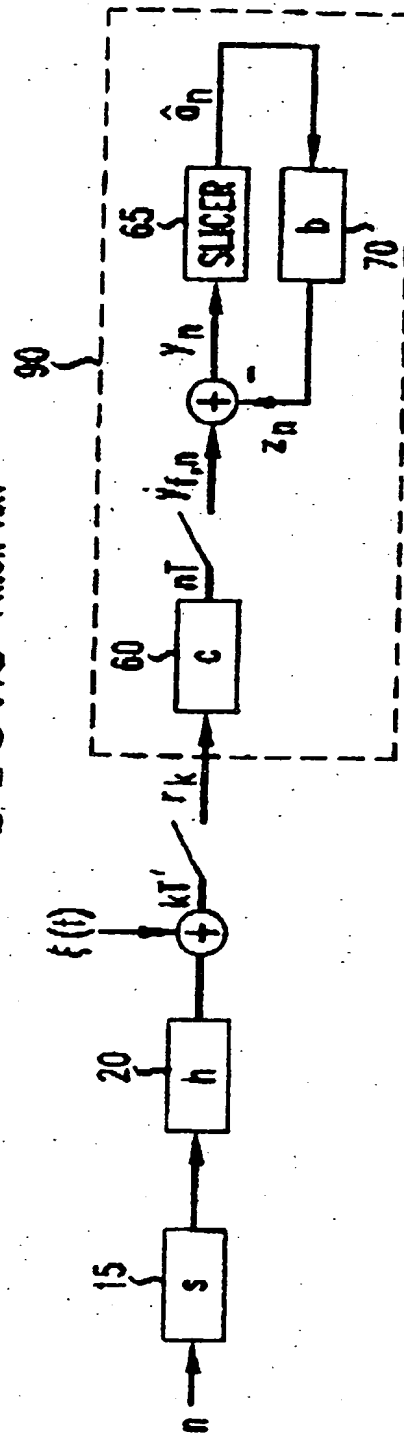


FIG. 3

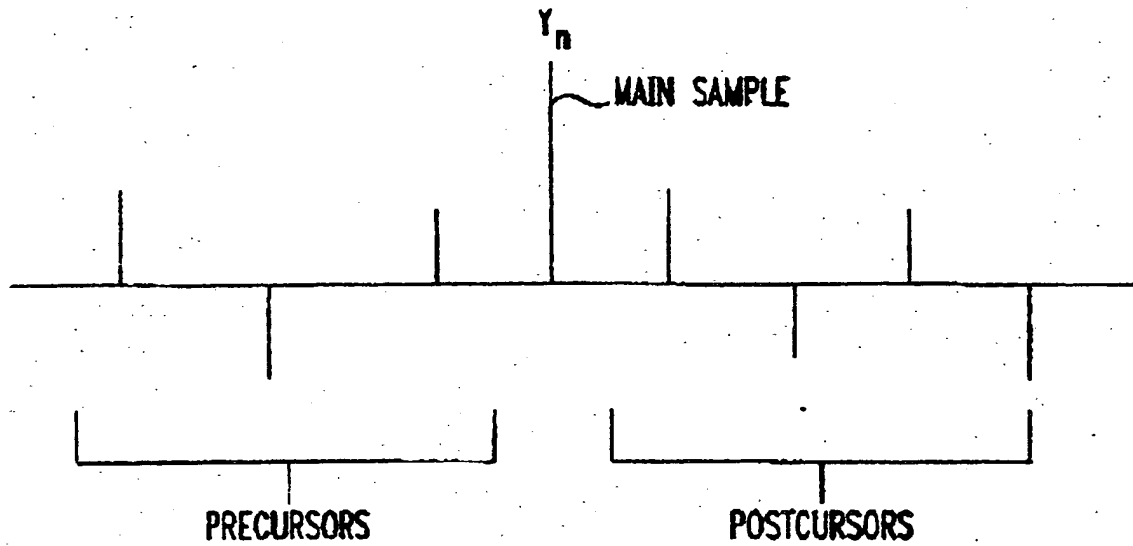


FIG. 6

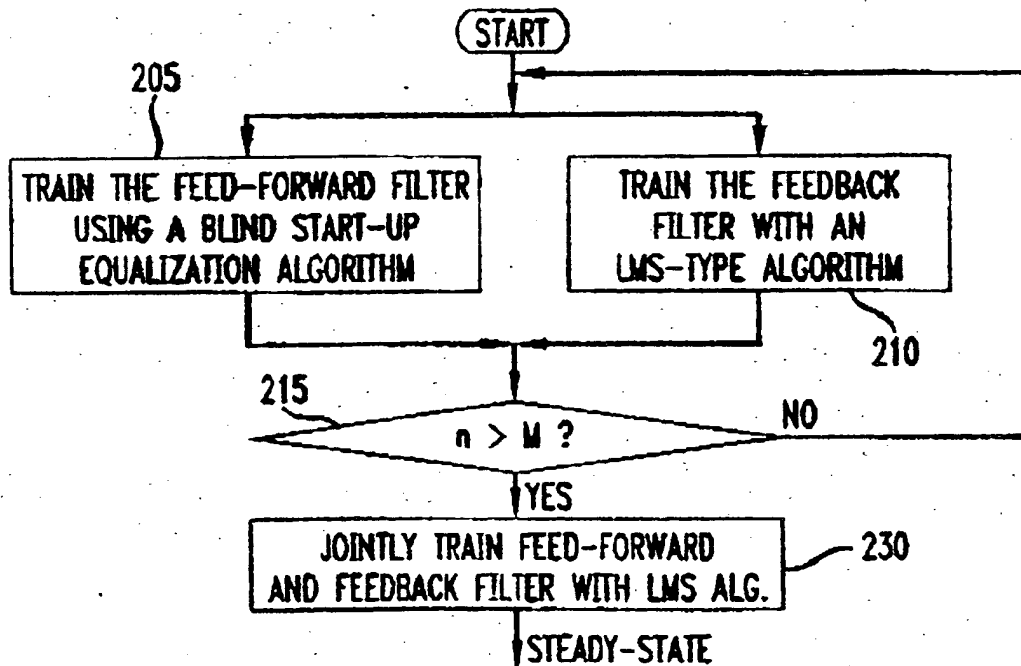


FIG. 4

PRIOR ART

100

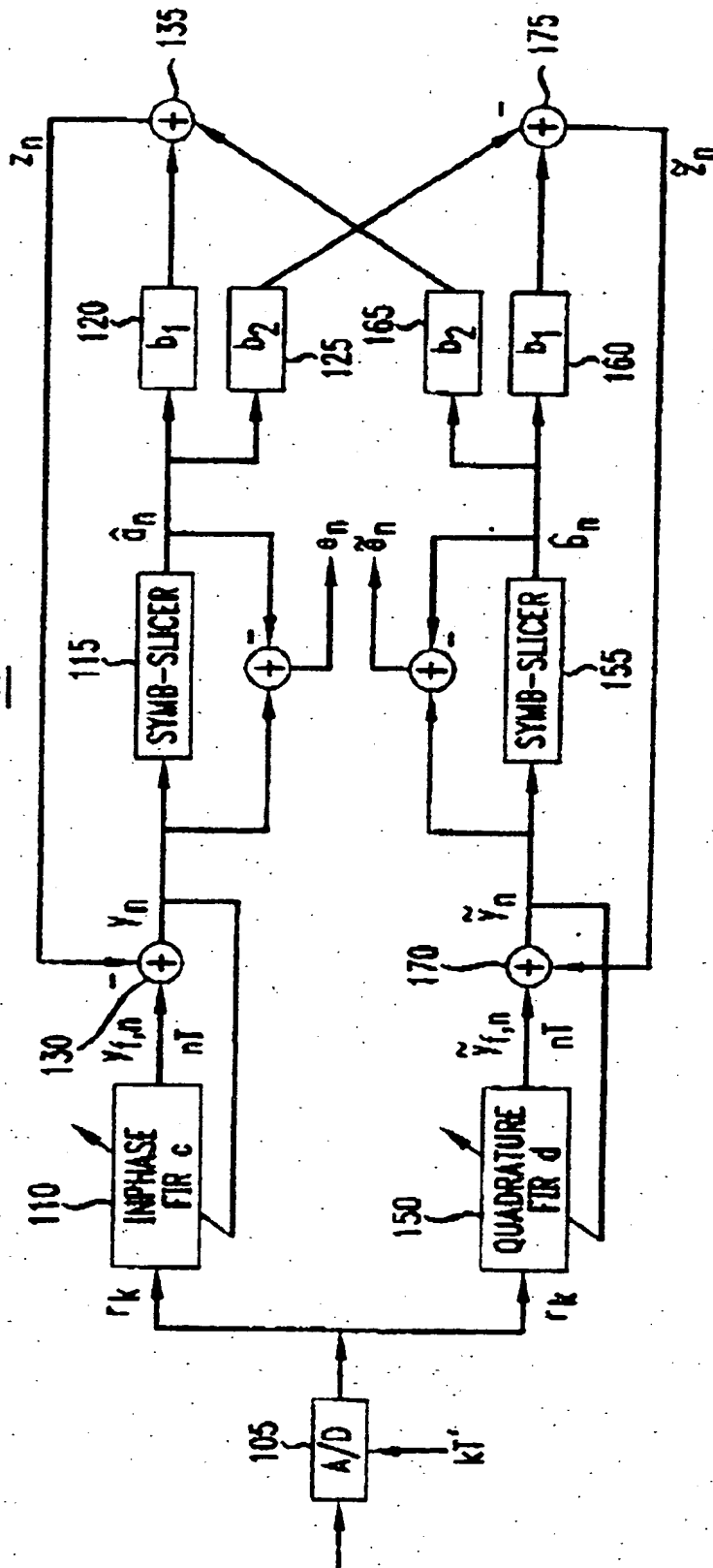


FIG. 5

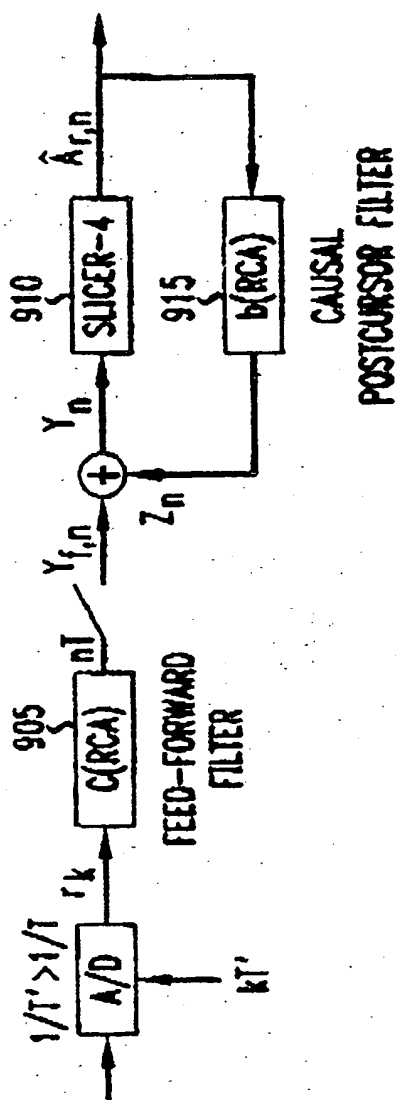


FIG. 9

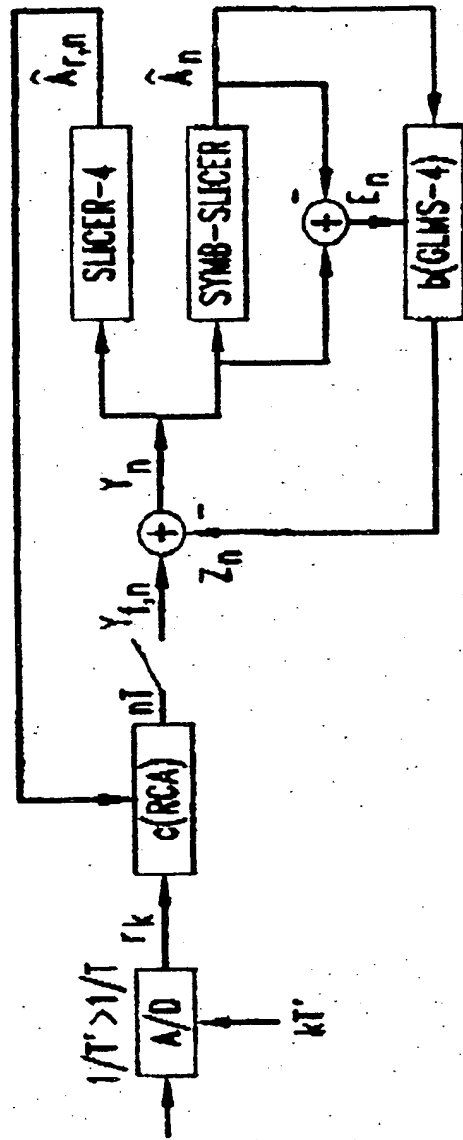


FIG. 10

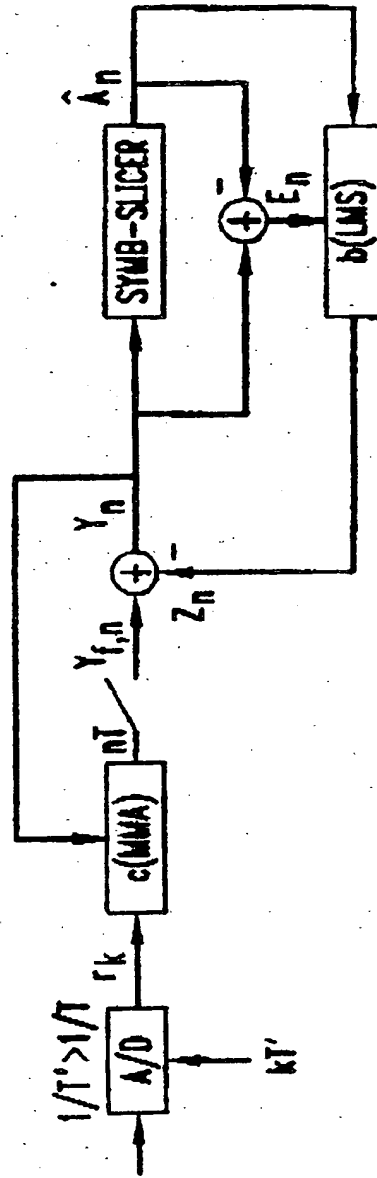




FIG. 12

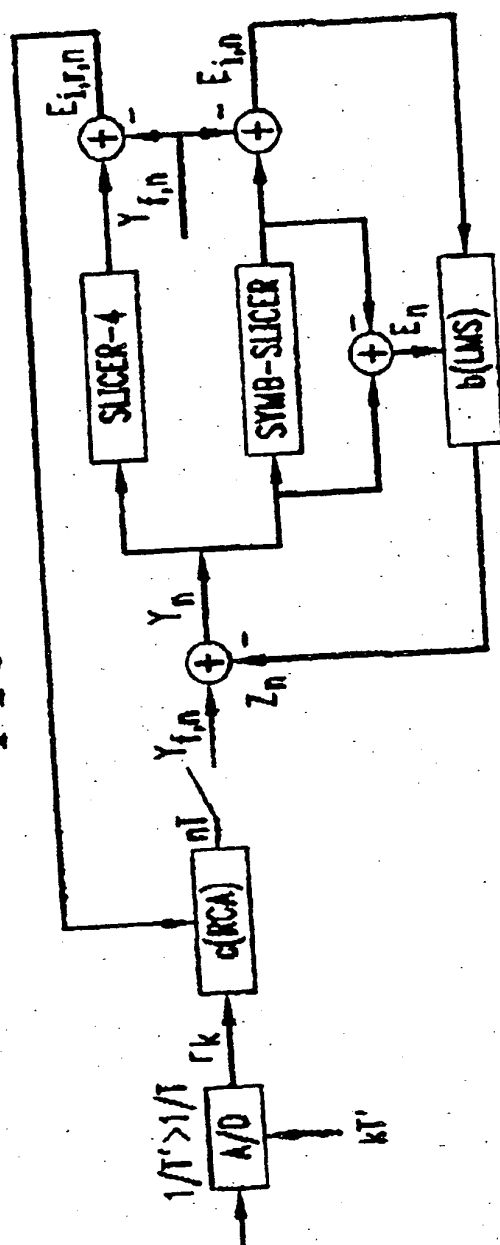


FIG. 13

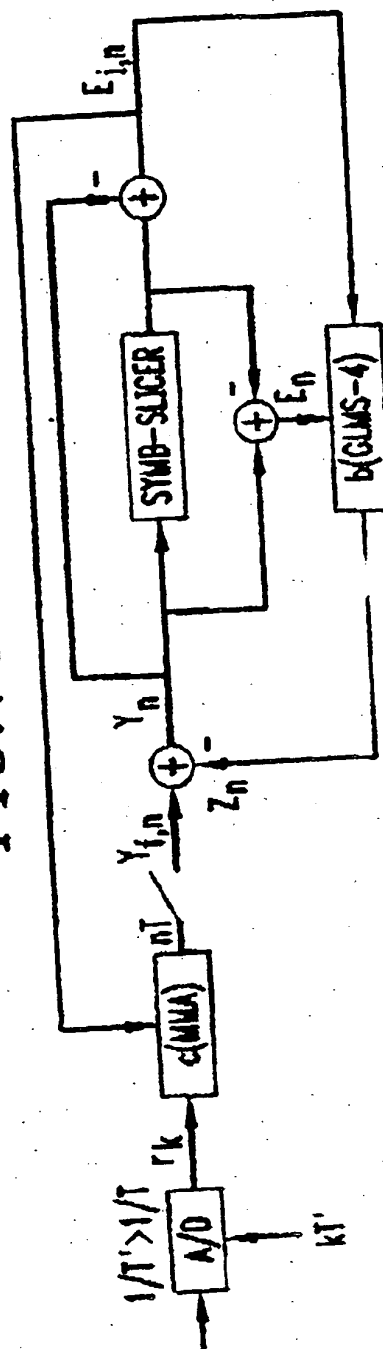
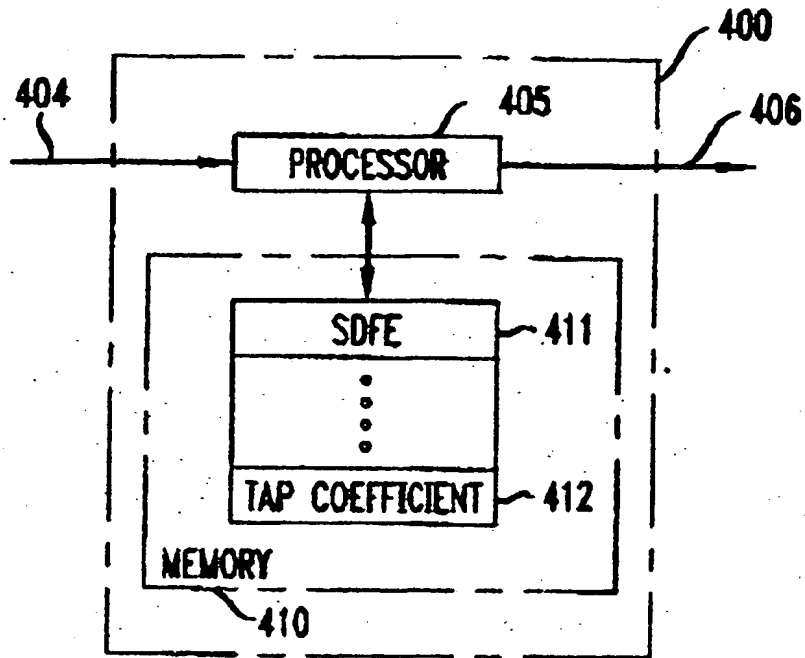
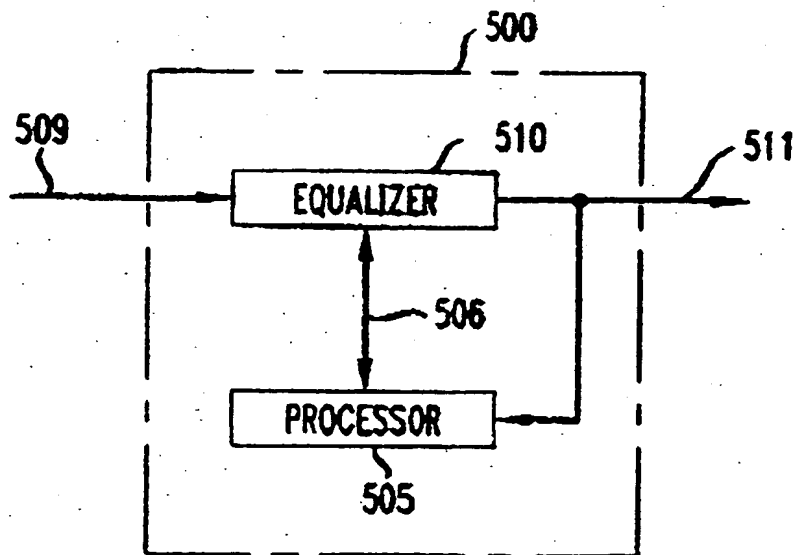


FIG. 14*FIG. 15*

THIS PAGE BLANK (USPTO)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☒ FADED TEXT OR DRAWING

☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☐ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)